Received August 6, 2017, accepted September 4, 2017, date of publication September 19, 2017, date of current version October 25, 2017.

*Digital Object Identifier 10.1109/ACCESS.2017.2752242*

# Optimizing Lifespan and Energy Consumption by Smart Meters in Green-Cloud-Based Smart Grids

**ISMA FARAH SIDDIQUI[1], SCOTT UK-JIN LEE[1], ASAD ABBAS[1], AND ALI KASHIF BASHIR[2], (Senior Member, IEEE)**

[1]Department of Computer Science and Engineering, Hanyang University ERICA, Ansan 15588, South Korea
[2]Department of Science and Technology, University of the Faroe Islands, Tórshavn FO100, Faroe Islands

Corresponding author: Scott Uk-Jin Lee (scottlee@hanyang.ac.kr)

This work was supported by the Research Fund of Hanyang University under Grant HY-2013-N.

**ABSTRACT** Green clouds optimally use energy resources in large-scale distributed computing environments. Large scale industries such as smart grids are adopting green cloud paradigm to optimize energy needs and to maximize lifespan of smart devices such as smart meters. Both, energy consumption and lifespan of smart meters are critical factors in smart grid applications where performance of these factors decreases with each cycle of grid operation such as record reading and dispatching to the edge nodes. Also, considering large-scale infrastructure of smart grid, replacing out-of-energy and faulty meters is not an economical solution. Therefore, to optimize the energy consumption and lifespan of smart meters, we present a knowledge-based usage strategy for smart meters in this paper. Our proposed scheme is novel and generates custom graph of smart meter tuple datasets and fetches the frequency of lifespan and energy consumption factors. Due to very large-scale dataset graphs, the said factors are fine-grained through R3F filter over modified Hungarian algorithm for smart grid repository. After receiving the exact status of usage, the grid places smart meters in logical partitions according to their utilization frequency. The experimental evaluation shows that the proposed approach enhances lifespan frequency of 100 smart meters by 72% and optimizes energy consumption at an overall percentile of 21% in the green cloud-based smart grid.

**INDEX TERMS** Green Cloud, Fog Computing, Smart Grid, IoT-enabled Smart Meter, Semantic Web.

## I. INTRODUCTION

Cloud computing is a robust platform that supports large-scale data processing in a parallel distributed environment [1]. It consists of a client-server architecture that includes services, protocols and infrastructure to perform remote tasks efficiently than traditional master-slave paradigm [2]. The cloud architecture is further categorized into multiple types i.e. location-based, service-based and environment-based cloud computing [3]. The accessibility-based cloud computing refers to the functional prototype that involves public, private, hybrid and community access of a client at a remote namespace. The service-based cloud computing involves features of subroutines i.e. IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service), SaaS (Software-as-a-Service) and storage functionality that delivers a service to the end user. The environment-based cloud computing consists of increasing efficiency, reducing energy consumption, enhancement in lifespan and optimization of application resources in a remote namespace [4].

The green cloud is an environment-based functionality that optimizes usage of resources and energy consumption in a remote namespace [5]. This approach is usually adopted by industries such as datacentres, manufacturing plants and smart grid infrastructures that adopts resource optimization techniques to reduce energy consumption and increase lifespan of devices in a parallel distributed computing environment [6]. The architecture of green cloud involves two types of master-slave communication i.e. (i) fog computing, and (ii) centralized computing. The fog computing consists of decentralized computing infrastructure, where data, computation, storage and applications are configured at most efficient and logical location between the cloud and data source. The centralized computing involves a compact architecture, where consistency is more important than effective placement of resources in a cloud [7].

The smart grid is a part of green cloud environment, where decentralized distribution units perform functional operations into a self-driven intelligent environment. The grid collaborates with multiple operational perspectives simultaneously i.e. interoperability, robust communication, intelligent agent, IoT sensors, IT infrastructure and its security, storage reservoirs, electric transportation and power systems [8] as shown in Fig. 1.

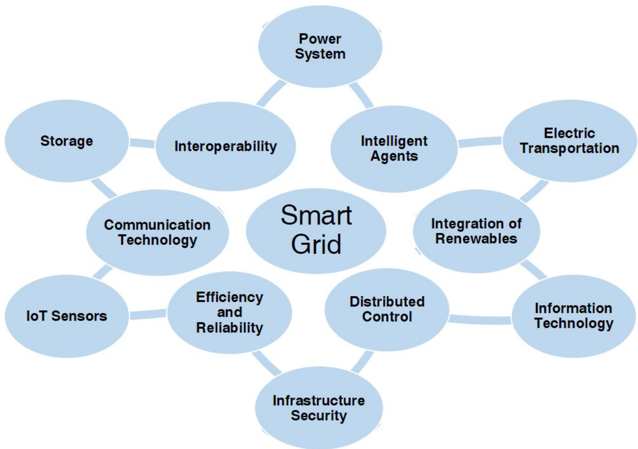


**FIGURE 1. Smart grid architecture.**

The grid controls the distributed units through software and hardware technologies. The hardware technology involves various IoT-enabled smart meters, sensor devices, power generation and distribution machinery, and power storage reservoirs [9]. The software technology includes semantic web applications, semantic reservoirs such as triplestore, and big data platform i.e. Hadoop that collects, manages, stores and performs analytics over information datasets in smart grid [10].

The smart meters record distribution unit information and bundle it over a node in tabular form [11]. The compiled bundles are then transformed to a grid readable format i.e. RDF [12]. The distribution nodes then process RDF datasets through transformation channels and store them into semantic reservoirs [11] as shown in Fig. 2.

Ideally, a smart meter is supposed to produce information records in a normal grid environment having an adequate length of recommended lifespan and usage of optimized energy consumption. But, in reality, a smart meter suffers from environmental issues and results in decreasing life span unusually along with abnormal routine of energy consumption [13]. The factors affecting lifespan of a smart meter consists of life expectancy (LE), genetics (GE), environment factors (EF), change over time (CT) and limited longevity (LL) [14]. By default, the factors affecting lifespan decreases with a normal range and a definite passage of time. Based on this assumption, production companies state lifespan decrement based on regular unit tests. But, in the real environment, factors do not decrease as expected in unit tests and depicts an unpredictable variation in ranges. Therefore, lifespan of a smart meter decreases rigorously

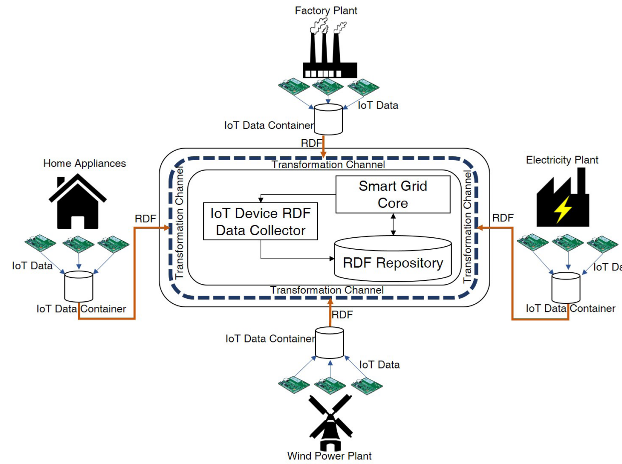


**FIGURE 2. Smart meters RDF data storage into smart grid.**

than proposed decrement and affects the operations of distribution node. In the same way, the factors affecting energy consumption involves voltage rise up (RU), maximum current (MC), voltage drop (VD), voltage observation (VO) and average voltage (AV). By default, the factors affecting energy consumption performs in a normal situation with routine parameters. But, in the real scenario, smart meters face voltage fluctuations more often and observes an unpredictable supply of voltage and current to the power component. As a result, it records abnormal readings of factors affecting energy consumption and produces malfunctioned information records [15].

After a smart meter produces malfunctioned information records, it is proposed to refurbish or replace with a new one. But, if the problem is observed in a large scale, it would scale smart grid budget to an approachable limit [16].

In order to resolve discussed problem, we propose a novel knowledge-based technique, that uses semantic reservoirs datasets and perform analytics to increase lifespan and optimize energy consumption. The presented approach formulates a custom RDF graph named as "Personalize graph", which fetches only factors' tuples from large-scale RDF datasets. After the fine-grained collection of RDF tuples, datasets are observed with a constraint of disorder range through Hungarian algorithm and filter out the smart meters decreasing life span and consuming abnormal range of energy. The result set is then shifted to logical partitions, where a smart meter would observe a workload as per its decreasing life span and energy consumption. As a result, the whole process increases the life span and optimize energy consumption of a smart meter in green-cloud-based smart grid.

The main contributions of this paper can be highlighted as follows.

- A novel RDF-based personalize graph to filter out factor dataset of lifespan and energy consumption.
- A novel knowledge-based error analytics in to extraction of tuple dataset.

- A novel approach of data acquisition zones i.e. Peak Block and Off Block.
- A novel approach of using enhanced Hungarian algorithm that identifies factor range of maximum usage of lifespan and energy consumption over RDF-based personalize graphs.
- A novel approach of logical partitions that put condition-based workload for extending lifespan and performing optimal energy consumption into smart meters.

The remaining paper is organized as follows. Section II highlights the brief overview and background of smart grid, smart meters and Semantic Web technologies. Section III discusses proposed approach in detail. Section IV shows the experimental evaluation and discuss the results in accordance with proposed approach. Finally, Section V presents conclusion and future works.

## II. OVERVIEW

In this section, we present a brief overview about smart grid, smart meters and semantic web technologies.

### A. SMART GRID

The smart grid is being considered as "The Next Generation Electricity Supply" for the future perspectives and has emerged as a convergence of information technology along with power system engineering and communication technologies [17]. The idea behind smart grid subjects to the enhancement of regular electricity grid with intelligence technologies [18]. The intelligent technology consists of digitization of the grid with two-way wired and wireless communication channels i.e. WiMax, Wifi, fibre optics etc [19]. The distribution units use smart meters to collect and store record information into storage nodes [20]. The information records are stored into semantic-aware reservoirs and assist the smart grid to perform analytics [21]. Smart grid supports self-healing functionality, which helps to deal failures and blackouts during recording and storing smart meter information in semantic reservoirs [22], [23].

### B. SMART METERS

The smart meters are used as basic recording devices of distribution units [24]. The meters use two-way communication between itself and the smart grid [25]. It manages sensor information into in-built RAM with a maximum limit of 1.3 million records and releases it into tabular form over distribution node [26]. The smart meter is programmed to work in multiple shifts i.e. peak time and off time. The peak time includes the maximum resource usage time line, where as, off time involves normal resource usage timeline [27]. The smart meter has a lifespan of 5 to 7 years on normal workload having ideal environment [28], [29].

### C. SEMANTIC WEB TECHNOLOGIES

The semantic web is an extension of the existing world wide web, where information is described in a meaningful format [30]. Semantic web consists of ontology, schema, Internationalized Resource Identifiers (IRIs) and service discovery languages. The frameworks that support processing datasets include Resource Description Framework (RDF), RDF Schema (RDFS), Simple Knowledge Organization System (SKOS), SPARQL Protocol and RDF Query Language (SPARQL), Notation3 (N3), N- Triples, Terse RDF Triple Language (Turtle), Web Ontology Language (OWL) and Rule Interchange Format (RIF) [31]. RDF framework is considered as a key construct to process semantic information through communication channels and store into semantic-aware reservoirs e.g., SESAME, AllegroGraph, Oracle11g and Jena-TDB [32]. RDF statement is known as triple and consists of three elements i.e. a subject, a predicate and an object. The literal/IRI supports to represent each element and the relationships [33], [34].

## III. PROPOSED APPROACH

The proposed approach involves operational steps to enhance lifespan and optimize energy consumption i.e. i) smart grid repository, ii) RDF-based personalize graphs, iii) approximation analytics over lifespan and energy consumption tuple datasets, iv) error factor in Personalize tuple dataset, v) enhancement in lifespan of smart meter, vi) optimization in energy consumption of smart meter, and finally vii) smart meter's partition-based allocation in smart grid.

At first, the presented approach generates RDF-based personalize graphs and then fetches the factors dataset that affects the lifespan and energy consumption of smart meters. The factors are then filtered out with enhanced Hungarian algorithm and affected smart meters are recommended to be placed at placement slots for optimal performance in smart grid.

### A. SMART GRID REPOSITORY

The smart grid repository is configured with functionality of Jena-TDB triplestore and contains record information of distribution node i.e. RDF tuple in it as shown in Fig. 3.

Let $D_n$ be an RDF Dataset having triplestore $RS = Store\{triple\,(s, p, o)\}$ as a resident entity of smart grid repository $SG_{repository}$ where s is subject, o is object and p is predicate. The notion can be represented as (x rdf:type ex:property) and the triple can be defined as,

$$triple\,(s, p, o) \in \;\; t = (I \cup N) \times I \times (I \times N \cup L_i) \tag{1}$$

where I is IRI, N is number of nodes and L is number of literals.

let G be the default graph with a finite set of source nodes $Sn$ having $\Pi$ as a suitable property in the dataset. Therefore, graph G can be represented a product set as,

$$G = \underbrace{Sn \times \Pi_1 \times ... \times \Pi_{i-1}}_{\text{B}} \times \underbrace{\Pi_i \times \ldots \times \Pi_{i+d}}_{\text{H}} \tag{2}$$

where $B = S \times O$ and $H = P$. The $RS$ is countable and keeps track of tuple data through indexes known as $idx$, where

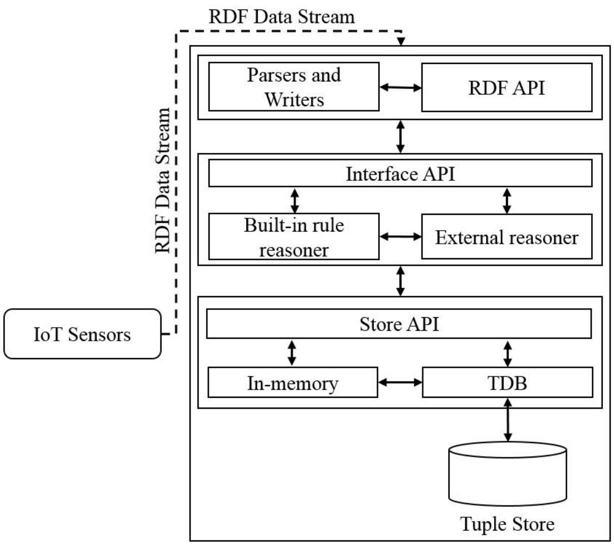


**FIGURE 3.** RDF tuple reservoir of smart grid.

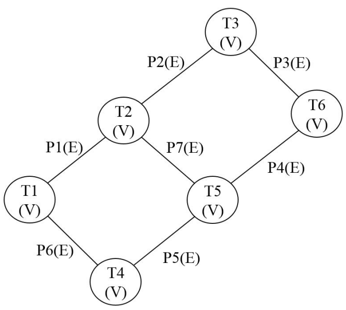


**FIGURE 4.** Basic union graph.

$idx_j : \Pi_j \rightarrow \mathbb{N}, j\epsilon [1, i+d]$. The structure of a dataset is represented as a union default graph as shown in Fig. 4, where V denotes a vertex and E represents an edge in the graph. The value of union graph can be represented as,

$$G_{Default} = union\,(G) \tag{3}$$

### B. RDF-BASED PERSONALIZE GRAPH

The smart grid consists of a huge semantic repository $SG_{repository}$, which stores tuple datasets of all the distribution nodes. By default, RDF graph database provides a querying technique that delivers custom extraction of tuples extracts [35]. However, smart grid deals with streaming RDF storage into $SG_{repository}$, therefore the query-based RDF graph extraction techniques would not be much feasible. The personalize RDF graph provides a custom approach for extracting dataset from large-scale streaming storage $SG_{repository}$ of smart grid.

A personalize graph is a graph $G = (V_p, E_i, L_m)$, where $V_p$ is a set of personalize vertices, $E_i$ a set of identical edges and $L_m$ relates to the mutual labeling function $\sum \mathbb{N} \times (V_p \cup E_i)$ between vertices $V_p$ and edges $E_i$ having index $idx_j\epsilon\mathbb{N}$. It consists of a collection of sub-graphs with specific mutual labeling functions to identify the label of a vertex or edge through index $idx_j$.

**Algorithm 1** RDF-Based Personalize Graph Generation for Custom Extraction of Tuples From $SG_{repository}$

**Input** $SG_{repository}$ having triplestore $RS$, a set of extraction depth $d$ and instance $I$.
**Output** A $SG$, personalized graph $G = (V_p, E_i, L_m)$ having mapping $V_n$ over integer and $E_n$ edge to integer for instance.
**procedure**
  **1. Initialize**
  **for** each $n \in I$: **do**
    add a vertex $m$ to $V_p$ having $L_m(m, d) = \&$ and $Link(i) = m$
  **2. Subgraph Procedure**
  **for** each $n \in I$: **do**
    Lookup $= n$
  **for** each $j = d - 1$ to $o$ : **do**
    newlookup $= \lambda$
    **for** each $r \in$ lookup: **do**
      $triples = RS$
      $RS = Store\{triple\ (s, p, o)\} \in triples$
      add $o$ to newlookup
      **if** Link($o$) is undefined **then**
        add vertex n to $V_p$ and
        Set Link($o$) = n
      set $L_m(Link(o), j) = o$
      **if** $V_n(Link(o))$ is undefined **then**
        set $V_n(Link(o)) = j$
      **if** $Link(RS)$ is undefined **then**
        add edge $g$ to $E_i$
        set $Link(RS) = g$
      set $L_m(Link(RS), j) = p$
      **if** $\&_i(Link(RS))$ is undefined **then**
        set $\&_i(Link(RS) = j$
      Newlookup $= j(Subgraph_{personalized})$
  Lookup = Newlookup

The sub-graph's vertices and edges identify each other through a property named "neighborhood". It completes the process of identification through a base neighborhood and edge neighborhood. The base neighborhood $N(v_p) = (v_p{}', v)\epsilon E_i$ consists of personalize edges $E_p$ and vertices $V_p$ and the edge neighborhood $N(v_p', v)$ represents a point, where edges join personalize vertices.

Algorithm-1 generates a personalize graph to extract subgraph with a depth $d$ from dataset $D_i$ over instance $i$ where $i\epsilon I$ (set of instances). Finally, mutual label $L_m$ joins all relevant sub-graphs and produces personalize graph $G$.

To this extend, the proposed approach has constructed a RDF-based personalize graph for smart grid triplestore. The next step involves integration of graph $G$ with streaming function that deals to the continuous buildup of stream-based RDF dataset into triplestore $SG_{repository}$.

**Algorithm 2** Identical Placement of Labeled Personalize Graph $G$

**Input** A personalize RDF graph $G = (V_p, E_i, L_m)$, sub-graph of depth $d$ and iteration $b$.
**Output** Personalize Labeling of graph $G$ having instances $(b_o \rightarrow b_m)$ with dictionary element $z$
**procedure**
  **for** $y = o$ to $m$ **do**
    **1. Label identification over multiset**
    **for** $g \in E_i$, $v \in V_p$ and $j = 0$ to $d$ **do**
      **if** $L_m(v, j)$ is defined and $y = 0$ **then**
        set $W_y(v, j) = b_o(v, j) = L_m(v, j)$
      **if** $L_m(g, j)$ is defined and $y = 0$ **then**
        set $W_y(g, j) = b_o(g, j) = L_m(g, j)$
      **if** $L_m(g, j)$ is defined and $y > 0$ **then**
        set $W_y(g, j) = \{L_{m-1}(t, j+1)/t \in N(e)\}$
      **if** $L_m(v, j)$ is defined and $y0$ **then**
        set $W_y(g, j) = \{L_{m-1}(t, j)/t \in N(v)\}$
    **2. Sort Label Identification**
    **for** each $W_y(v, j)$, $W_y(g, j)$ **do**
      sort $W_y(v, j)$, $W_y(g, j)$ in ASC order
      concatenate $\rightarrow S_y(v, j)$ and $S_y(g, j)$
    **for** each $S_y(v, j)$ and $S_y(g, j)$ **do**
      **if** $y > 0$ **then**
        Add $L_m(v, j), L_{m-1}(g, j) \rightarrow$ prefix($S_y(v, j)$ and $S_y(g, j)$)
    **3. Personalize Label Compression**
    **for** each $S_y(v, j)$ and $S_y(g, j)$ **do**
      Link $S_y(v, j)$ , $S_y(g, j)$ to new personalized label with function
      $f\colon \sum_p^{\infty} \rightarrow \sum, f(S_y(v, j)) = f(S_y(v', j))$ iff $S_y(v, j) = S_y(v', j)$ and $f(S_y(g, j)) = f(S_y(g', j))$ iff $S_y(g, j) = S_y(g', j)$
    **4. Personalized Labeling**
    **for** each $S_y(v, j)$ and $S_y(g, j)$ **do**
      set $L_m(v, j) = f(S_y(v, j))$ and $L_m(g, j) = f(S_y(g, j))$

Let $G$ is the personalize graph that extracts custom dataset from reservoir $SG_{repository}$ and proposes a method to maintain a continuous tuple-flow using identical placement of label $b$ iterations, where a single label appears between instances $L_{m_{o \rightarrow b}}$. The personalize graph $G$ between two instances can be computed as,

$$G_p = \sum_{n=0}^{b} \frac{n+1}{b+1} G_{RDF}\left((V_p, E_i)\left(V_{p'}, E_{i'}\right)\right) \tag{4}$$

where,

$$G_{RDF}\left((V_p, E_i)\left(V_{p'}, E_{i'}\right)\right) = \sum_{(v,d)\epsilon V_p} \sum_{(v',d')\epsilon V_{p'}} \delta\left(L_m(v, d), L_m(v', d')\right) + \sum_{(E,d)\epsilon E_i} \sum_{(E',d')\epsilon E_{i'}} \delta\left(L_m(E, d), L_m(E', d')\right)$$

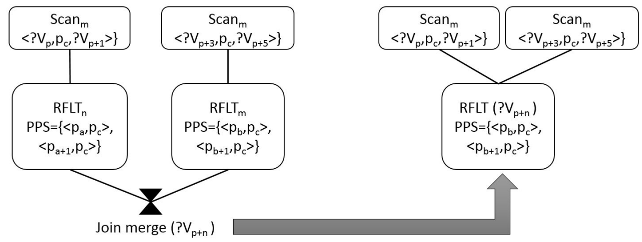


**FIGURE 5.** Personalize RDF filter for graph $G$.

The $\delta$ is the Dirac absolute for testing equality among edges and vertices of sub-graphs [36]–[38]. Algorithm-2 initially identifies labels of multiple sub-graph sets and re-configures them to acquire personalize labeling. In the second step, compression takes place to adjust a binary format with personalize graph and dispatches labeled sub-graphs for personalize graph $G$. After, the proposed approach formulated a general purpose technique of personalize graph $G$, it is applied over smart meter's lifespan and energy consumption use cases discussed below.

Let $G_{Lifespan}$ be a personalize graph to extract smart meter's activity dataset from smart grid reservoir $SG_{repository}$. The $RS$ tuple dataset uses a filtering technique named RDF Triple Filtering (R3F) [39], [40], which helps graph $G_{Lifespan}$ to build lifespan sub-graph index matching elements though $idx_j$ through Vertex $V_p$ and Edges $E_i$. The tuple filtering method is carried with sorted $idx_j$ as,

$$G_{Lifespan}(Sort\left(idx_j\right)) = \frac{V_p \times E_i}{Scan_{idx_j}} \tag{5}$$

where $Sort$ method identifies sub-graphs attached to the vertices $V_p$ and $Scan_{idx_j}$ identifies predicates $p$ over index $idx_j$. The predicate path of graph $G_{Lifespan}$ is computed as,

$$G_{Lifespan}(Path) = (Scan_{idx_j}, G_{Lifespan}(Sort), maxL) \tag{6}$$

Where $maxL$ is the maximum path length. The predicate path set (PPS) collects predicates in coordination with $Scan_i$ array and formulate a two-dimension array through $idx_j$ and RFLT operator. The collection of two or more than two PPS merger through $Scan_{idx_j}$ is shown in Fig. 5 and elaborated as,

$$G(index) = \left(\sum_{p\epsilon PPS} \left(idx_j\,(p)\right)\right) \tag{7}$$

The scanning and searching of sub-graphs is shown in Fig. 6. The graph search $G(Search)$ of graph $G_{Lifespan}$ can be obtained as,

$$G(Search) = \left(\sum_{scan\epsilon subgraph} \left(Scan_{idx_j}\right)\right) + G\,(index) \tag{8}$$

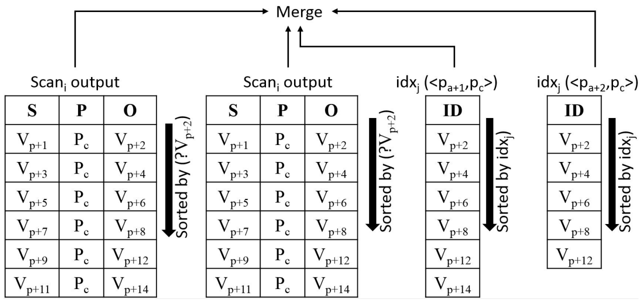


**FIGURE 6.** **Personalize RDF Filter to merge sub-graphs [39], [40].**

In order to obtain personalize tuple dataset of graph $G_{Lifespan}$, we use eq(4) and eq(8) as,

$$G_{Lifespan} = \left( \sum_{n=0}^{b} \frac{n+1}{b+1} G_{RDF} \left( (V_i, E_i) \, (V_{i'}, E_{i'}) \right) \right) + G(Search) \quad (9)$$

$G_{Lifespan}$ is a personalize graph and fetches a custom tuple dataset for lifespan values of smart meters. Similarly, we obtain personalize graph $G_{Energy}$ as,

$$G_{Enery} = \left( \sum_{n=0}^{b} \frac{n+1}{b+1} G_{RDF} \left( (V_i, E_i) \, (V_{i'}, E_{i'}) \right) \right) + G(Search) \quad (10)$$

### C. APPROXIMATION ANALYTICS OVER LIFESPAN AND ENERGY CONSUMPTION TUPLE DATASETS

By default, a smart meter consumes lifespan through a standard procedure [41] as,

$$min(\frac{b_i}{p_i}) \quad 1 \le i \le M(s) \quad (11)$$

where $b_i$ is battery energy, $p_i$ is power consumed and $M(s)$ represents smart meter. This consumption of lifespan is related with measurements through electro-mechanical operations. But, the proposed approach deals with knowledge-based measurement of lifespan, therefore, we focus to evaluate results through personalize graph datasets. The personalize graph $G_{Lifespan}$ performs triple extraction and gets dataset of smart meter as,

$$SM_{Lifespan} = SM_{TMD} - aggr(G_{Lifespan}) \quad (12)$$

where $SM_{TMD}$ represents total tuples of a smart meter, $aggr(G_{Lifespan})$ represents the method of filtering lifespan tuples from dataset.

Similarly, energy consumption of a smart meter can be carried out through,

$$SM_{Energy} = [(1 - \xi_i) + \xi_i d_i^2](\frac{J}{Message}) \quad (13)$$

where $\xi_i$ represents message emission probability to $SG_{repository}$, $d_i^2$ is the cost for emitting a message from smart meter. Since, this method is applicable when we deal with electro-mechanical procedures therefore, we adopt the analytics of tuple procedure and evaluates energy consumption through personalize graph tuple datasets. The graph $G_{Energy}$ performs tuple extraction of energy consumption and returns dataset of smart meter as,

$$SM_{Energy} = SM_{TEC} - aggr(G_{Energy}) \quad (14)$$

where $SM_{TEC}$ is tuple dataset of total energy consumption, and $aggr(G_{Energy})$ represents the method of filtering energy consumption tuples from dataset.

#### 1) SMART METER LOAD SHIFT

Smart grid activates meters in two shifts i.e. (i) Peak load and (ii) Off load [42]. The generation of tuples in peak load are stored into $SG_{repository}$ and represented as,

$$SG_{repository(PeakBlock)} = RS(dataset_{Peak-load}) \quad (15)$$

Similarly, the generation of tuples in off load are stored into $SG_{repository}$ and represented as,

$$SG_{repository(OffBlock)} = RS(dataset_{off-load}) \quad (16)$$

The ratio of Peak-to-Off tuple load can be calculated as,

$$Ratio_{(Peak-to-Off)} = \frac{SG_{repository(PeakBlock)}}{SG_{repository(OffBlock)}} \quad (17)$$

### D. ERROR FACTOR IN PERSONALIZE TUPLE DATASET

The extraction of personalize tuple datasets generates error exceptions due to unidentified edge, unknown vertex and array out of index issues in peak and off blocks. The error percentile can be obtained as,

$$Error_{Peak-to-Off} = Error_{Peak} + Error_{Off} \quad (18)$$

where $Error_{Peak}$ represents number of errors in peak load and $Error_{Off}$ depicts number of errors in off load dataset.

### E. ENHANCEMENT IN LIFESPAN OF SMART METER

The proposed approach enhances lifespan of smart meter through calculating remaining life $SM_{RL}$ as,

$$SM_{RL} = SM_{TL} - SM_{UL} \quad (19)$$

where $SM_{TL}$ represents total lifespan mentioned over device specification and $SM_{UL}$ is used lifespan, that is calculated through peak and off block dataset tuples of $G_{Lifespan}$. The factors affecting $SM_{RL}$ includes life expectancy $LE_f$, genetics $GE_f$, environmental factors $EF_f$, change over time $CT_f$ and limited longevity $LL_f$. The commutative factor $\Delta F$ can be represented as,

$$\Delta F_{Lifespan} = LE_f + GE_f + EF_f + CT_f + LL_f \quad (20)$$

The threshold point of the above factors can be determined through $\Delta T$ as,

$$\Delta T_{Lifespan} = LE_t + GE_t + EF_t + CT_t + LL_t \quad (21)$$

The maximum utilization factor of $\Delta T_{Lifespan}$ are found using Hungarian algorithm [43] as shown in Fig. 7.

| | *Peak* | *Off* |
|---|---|---|
| $LE_t$ | 0.11 | 0.07 |
| $GE_t$ | 0.19 | 0.11 |
| $EF_t$ | 0.21 | 0.13 |
| $CT_t$ | 0.09 | 0.05 |
| $LL_t$ | 0.25 | 0.16 |

| | *Peak* | *Off* |
|---|---|---|
| $LE_t$ | -0.11 | -0.07 |
| $GE_t$ | -0.19 | -0.11 |
| $EF_t$ | -0.21 | -0.13 |
| $CT_t$ | -0.09 | -0.05 |
| $LL_t$ | -0.25 | -0.16 |

| | *Peak* | *Off* |
|---|---|---|
| $LE_t$ | 0.14 | 0.18 |
| $GE_t$ | 0.06 | 0.14 |
| $EF_t$ | 0.04 | 0.12 |
| $CT_t$ | 0.16 | 0.20 |
| $LL_t$ | 0.00 | 0.09 |

| | *Peak* | *Off* |
|---|---|---|
| $LE_t$ | 0.14 | 0.09 |
| $GE_t$ | 0.06 | 0.05 |
| $EF_t$ | 0.04 | 0.03 |
| $CT_t$ | 0.16 | 0.11 |
| $LL_t$ | ~~0.00~~ | ~~0.00~~ |

| | *Peak* | *Off* |
|---|---|---|
| $LE_t$ | 0.11 | 0.06 |
| $GE_t$ | 0.03 | 0.02 |
| $EF_t$ | 0.01 | 0.00 |
| $CT_t$ | 0.13 | 0.08 |
| $LL_t$ | ~~0.00~~ | ~~0.00~~ |

| | *Peak* | *Off* |
|---|---|---|
| $LE_t$ | 0.11 | 0.07 |
| $GE_t$ | 0.19 | 0.11 |
| $EF_t$ | 0.21 | **0.04** |
| $CT_t$ | 0.09 | 0.05 |
| $LL_t$ | **0.08** | 0.16 |

**FIGURE 7.** **Minimum life span using Hungarian Algorithm.**

| | *Peak* | *Off* |
|---|---|---|
| $RU_t$ | 0.31 | 0.16 |
| $MC_t$ | 0.27 | 0.15 |
| $D_t$ | 0.35 | 0.19 |
| $O_t$ | 0.11 | 0.06 |
| $AV_t$ | 0.21 | 0.12 |

| | *Peak* | *Off* |
|---|---|---|
| $RU_t$ | -0.31 | -0.16 |
| $MC_t$ | -0.27 | -0.15 |
| $D_t$ | -0.35 | -0.19 |
| $O_t$ | -0.11 | -0.06 |
| $AV_t$ | -0.21 | -0.12 |

| | *Peak* | *Off* |
|---|---|---|
| $RU_t$ | 0.04 | 0.19 |
| $MC_t$ | 0.08 | 0.20 |
| $D_t$ | 0.00 | 0.16 |
| $O_t$ | 0.24 | 0.29 |
| $AV_t$ | 0.14 | 0.23 |

| | *Peak* | *Off* |
|---|---|---|
| $RU_t$ | 0.04 | 0.03 |
| $MC_t$ | 0.08 | 0.04 |
| $D_t$ | 0.00 | 0.00 |
| $O_t$ | 0.24 | 0.13 |
| $AV_t$ | 0.14 | 0.07 |

| | *Peak* | *Off* |
|---|---|---|
| $RU_t$ | 0.01 | 0.00 |
| $MC_t$ | 0.05 | 0.01 |
| ~~$D_t$~~ | ~~0.00~~ | ~~0.00~~ |
| $O_t$ | 0.21 | 0.10 |
| $AV_t$ | 0.11 | 0.04 |

| | *Peak* | *Off* |
|---|---|---|
| $RU_t$ | 0.31 | **0.26** |
| $MC_t$ | 0.27 | 0.15 |
| $D_t$ | **0.35** | 0.19 |
| $O_t$ | 0.11 | 0.06 |
| $AV_t$ | 0.21 | 0.12 |

**FIGURE 8.** **Maximum energy consumption using Hungarian Algorithm.**

### *F. OPTIMIZATION IN ENERGY CONSUMPTION OF SMART METER*

The proposed approach reduces energy consumption of smart meter through calculating remaining energy cycles $SM_{RE}$ as,

$$SM_{RE} = SM_E - SM_{UE} \tag{22}$$

where $SM_E$ represents total cycles of energy consumption over device specification and $SM_{UE}$ depicts used energy cycles, that can be calculated through peak and off blocks dataset tuple of $G_{Energy}$. The factors affecting $SM_{RE}$ includes rise up voltage $RU_f$, maximum current $MC_f$, voltage drop $D_f$, voltage observation $O_f$ and average voltage $AV_f$. The commutative factor $\Delta F$ can be represented as,

$$\Delta F_{Energy} = RU_f + MC_f + D_f + O_f + AV_f \tag{23}$$

The threshold point of the above factors can be determined through $\Delta T$ as,

$$\Delta T_{Energy} = RU_t + MC_t + D_t + O_t + AV_t \tag{24}$$

The maximum utilization factor of $\Delta T_{energy}$ are found using Hungarian algorithm as shown in Fig. 8.

### *G. SMART METER' PARTITION-BASED ALLOCATION IN SMART GRID*

As per the obtained phase of knowledge-based analytics, we find that a smart meter works over a principle defined as: $\phi\left(\left(L_p, L_o\right)\left(E_p, E_o\right)\right)$, where $\left(L_p, L_o\right)$ represents maximum lifespan values for peak and off tuple dataset and $\left(E_p, E_o\right)$ depicts maximum energy consumption values for peak and off tuple dataset as shown in Fig. 9. The proposed approach takes advantage of this analytics and re-allocates smart meters with respect to their calculated factors of $\Delta T_{Lifespan}$ and $\Delta T_{Energy}$. After evaluating the status of smart meter, it is placed into a recommended usage slot. The configurations of each slot are discussed below.

- If the smart meter factors are consumed in-between 0 to 30%, it is declared as 'FULL' and places into slot 'A'. This slot uses smart meters at their full capacity into peak and off blocks.


**FIGURE 9.** **Default smart meter placement into smart grid.**

- If the smart meter factors are consumed in-between 30% to 50%, it is declared as 'NOT FULL' and places into slot 'B'. This slot uses smart meters at $\frac{1}{2}$ capacity into peak and off blocks.
- If the smart meter factors are consumed in-between 50% to 70%, it is declared as 'HALF' and places into slot 'C'. This slot uses smart meters at $\frac{1}{3}$ capacity into peak and off blocks.
- If the smart meter factors are consumed in-between 70% to 95%, it is declared as 'NOT HALF' and places into slot 'D'. This slot uses smart meters at $\frac{1}{6}$ capacity into peak and off blocks as shown in Fig. 10.

## IV. PERFORMANCE EVALUATION

The proposed approach is evaluated through smart meter tuple dataset stored in semantic triplestore Jena-TDB [44]. The evaluation includes, (i) generation of personalize graph, (ii) Tuple dataset of peak load and off load blocks, (iii) error ratio percentile in personalize graph, (iv) threshold extraction of factors affecting lifespan and energy consumption, and finally (v) the placement of smart meters in respective slot for optimal utilization in smart grid.

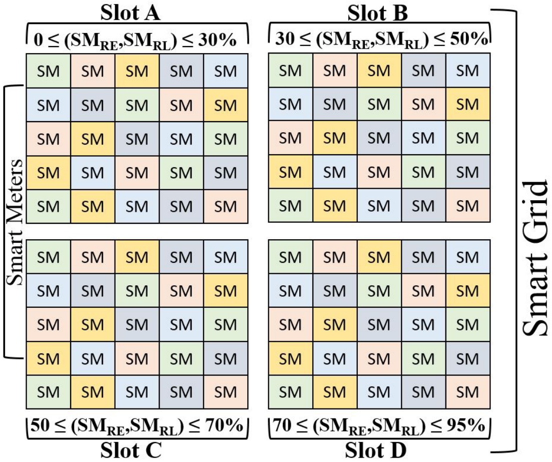


**FIGURE 10.** Partition-based placement of smart meters into smart grid.

### A. TESTBED

The testbed includes a configuration of Jena-TDB version 3.0.1 repository on Intel Core(TM) i5-3470 CPU @ 3.20 GHz having 6GM RAM with Windows 8.1 64bit OS having X64-based processor over 250GB WD Hard disk.

### B. SMART METER DATASET

We use synthetic workload of smart meters having distributed node tuple dataset [45]. The configuration of Jena-TDB is set to 300 GB tuple dataset workload with configuration of workloads i.e. (i) peak load and (ii) off load. The peak load consists of regular time usage while off load involves passive time usage of smart meters. We use multiple datasets having peak and off workloads of 20GB, 50GB, 100GB and 150GB datasets. We use Hadoop ecosystem [46] to perform analytics of lifespan and energy consumption factors and presents a scenario of giant reservoir of smart grid.

### C. PERSONALIZE GRAPH GENERATION

The personalize graph generation procedure builds a stack of custom queries that fetches tuples from Jena-TDB repository. The custom queries are divided into two types (i) static query and (ii) dynamic query. The static query adopts the functionality of personalize graph for fetching the tuples related to lifespan and energy consumption and dynamic query builds an array to pile up the continuous tuple flow and builds an array for input of static query. As we know that, $\Delta T_{Lifespan}$ includes 5 types of tuple factor, therefore, the proposed approach adopts 6 tuple extractor instances over lifespan dataset. In the same way, $\Delta T_{Energy}$ also includes 5 types of tuple factor, hence, the proposed approach also adopts 6 tuple extractor instances over energy consumption dataset. Thus, a personalize graph uses 12 tuple extractor instances for obtaining custom tuple dataset. We perform tuple extraction through the proposed personalize graph, union graph and simple graph over $SG_{Repository}$ dataset having size of 20 GB. The basic and union graph extractions are much faster than personalize graph because, they use full schema of RDF, where as personalize graph extracts custom tuple dataset with conditional attribute of fetching lifespan and energy consumption tuple dataset. We find that, personalize graph fetches tuple datasets at '19324' seconds over peak block as shown in Fig. 11. In the same way, personalize graph fetches tuple datasets at '11490' seconds over off block as shown in Fig. 12.

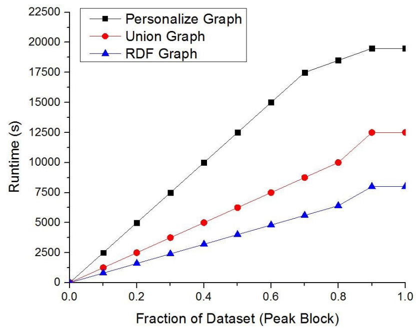


**FIGURE 11.** Personalized graph generation in peak-block SG repository.

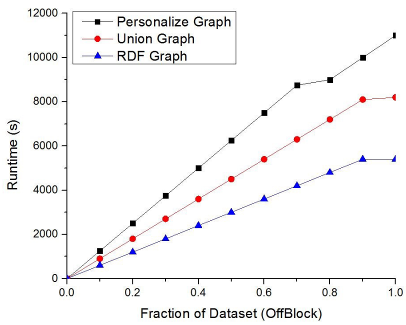


**FIGURE 12.** Personalized graph generation in off-block SG repository.

### D. SMART METERS IN PEAK-TO-OFF BLOCK TUPLE DATASET

The personalize graph extracts unstructured dataset that contains unordered tuple sets of smart meters. Hence, it increases complexity to identify number of smart meters used to produce semantic tuples into such a huge dataset. For this purpose, we use label strategy of personalize graph and take shifts into account i.e. peak block and off block and perform evaluation to extract number of smart meters from the tuple dataset. We find that, peak block contains 56% of in-use smart meters, where as, off block contains 44% of in-use smart meters in the 20GB tuple dataset as shown in Fig. 13. Similarly, we evaluate that peak block contains 59% of in-use smart meters, where as, off block contains 41% of in-use smart meters in the 50GB tuple dataset as shown in Fig. 14. In the same way, we observe that peak block contains 57% of in-use smart meters, where as, off block contains 43% of in-use smart meters in the 100GB tuple dataset as shown in Fig. 15. And, we find that peak block contains 64% of in-use smart meters, where as, off block contains 36% of in-use smart meters in the 150GB tuple dataset as shown in Fig. 16. The overall usage of smart meters in respective blocks shows an average usage of 62% smart meters in peak block and 38% utilization in off blocks of smart grid.

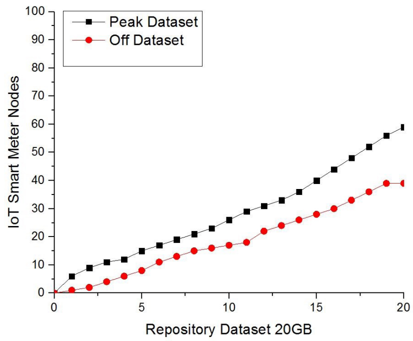


**FIGURE 13.** Smart meter nodes in 20GB repository.

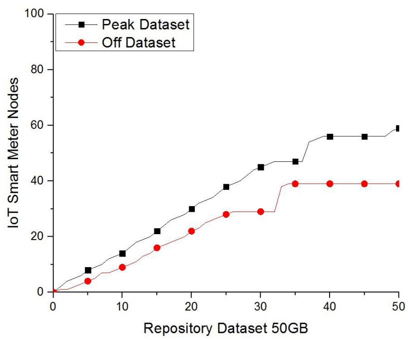


**FIGURE 14.** Smart meter nodes in 50GB repository.

### E. ERROR RATIO PERCENTILE IN PERSONALIZE GRAPH

The tuple extraction in personalize graph causes error exceptions due to unidentified edge, unknown vertex and array out of index problems. This happens due to improper insertion of tuple value i.e. value# and returns an unrecognizable format tuple. As a result, sub-graphs skip label connectivity and resume the graph with next element of the token. The personalize graph extracts tuple dataset from 20GB, 50GB, 100GB and 150GB and maintains a log at a user defined path. The error ratio of peak-to-off block is evaluated over datasets individually and observed error percentile of 2% over 20GB, 3.5% over 50GB, 5% over 100GB and 7.5% over 150GB respectively as shown in Fig. 17.

### F. THRESHOLD EXTRACTION OF FACTORS AFFECTING LIFESPAN AND ENERGY CONSUMPTION

The peak and off block datasets consist of multiple factor values obtained through personalize graph $G_{Lifespan}$ and $G_{Energy}$. The factor tuples acquire an adequate range of values that are further passed over a filter of Hungarian algorithm and generates a sequence of smart meters that has consumed lifespan and energy in descending order. We evaluate peak

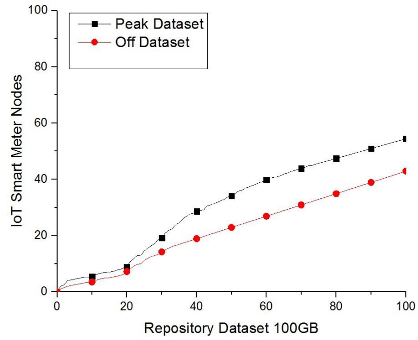


**FIGURE 15.** Smart meter nodes in 100GB repository.

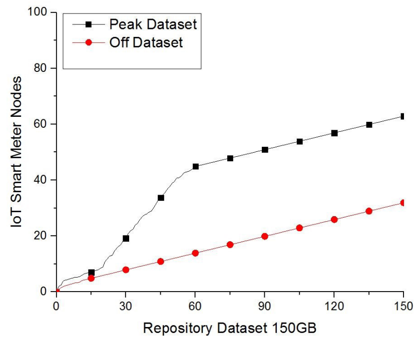


**FIGURE 16.** Smart meter nodes in 150GB repository.

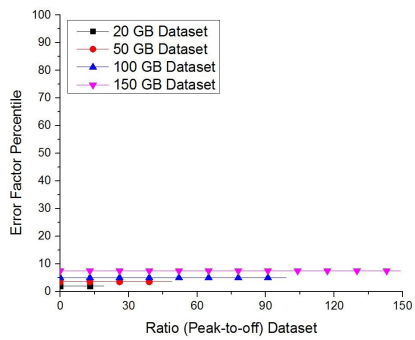


**FIGURE 17.** Error ratio in personalize graph.

and off tuple datasets of lifespan through discussed filter and find that, peak dataset contains 30 smart meters, and have consumed 11% LE, 12% GE, 14% EF, 19% CT and 3% LL averagely as shown in Fig. 18. In the same way, we find peak and off tuple datasets through Hungarian algorithm filter and evaluate that, off dataset contains 30 smart meters, and have consumed 9% LE, 11.7% GE, 13.8% EF, 18.6% CT and 2.9% LL averagely as shown in Fig. 19. Moreover, we find peak tuple datasets of energy through Hungarian filter and find that, off dataset contains 100 smart meters, and have consumed 2% RU, 12% MC, 14% VD, 19% O and 64% AV averagely as shown in Fig. 20. In the last, we evaluate off tuple datasets of energy through the same Hungarian filter

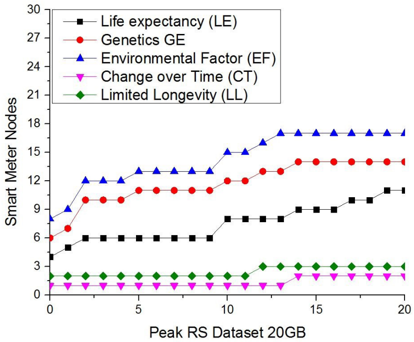


**FIGURE 18.** Factors affecting lifespan in peak block.

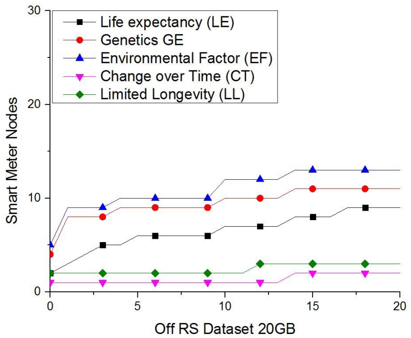


**FIGURE 19.** Factors affecting lifespan in off-block.

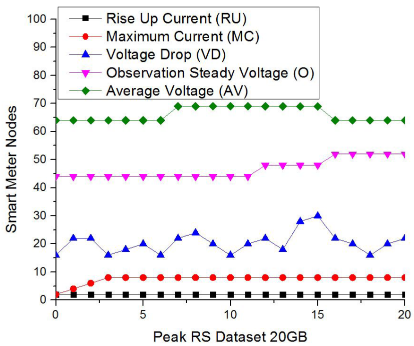


**FIGURE 20.** Factors affecting energy utilization in peak-block.

and find that, off dataset contains 100 smart meters, and have consumed 1.9% RU, 11.8% MC, 13.9% VD, 18.8% O and 60% AV averagely as shown in Fig. 21.

### *G. SMART METERS PARTITION-BASED PLACEMENT*

The slot configuration of smart meter consists of a controller that manages the recording requests of distribution nodes and instruct smart meters to perform operations with in a defined range of time. It also controls configuration script that generate number of records per cycle by a smart meter. In this way, a slot can effectively use a smart meter as per given optimization configuration and enhance the lifespan with minimum usage of energy in the smart grid. After performing the above experimental evaluations, we find that as per the identification of malfunctioned smart meters from peak

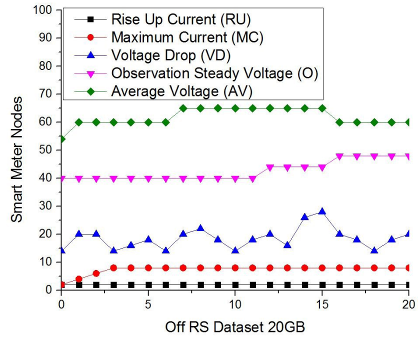


**FIGURE 21.** Factors affecting energy utilization in off-block.

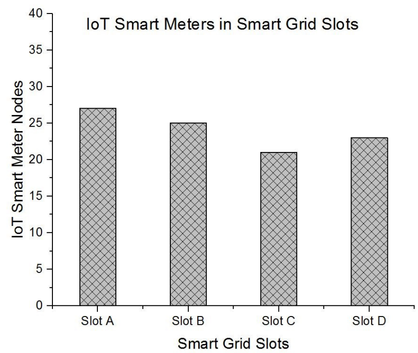


**FIGURE 22.** Smart meters allocation to partition-based slots.

and off block tuple datasets, the proposed approach places '27' meters in slot 'A', '25' meters in slot 'B', 21 meters in slot 'C' and '23' meters in slot 'D'. The remaining '4' unmarked smart meters have consumed their capacity beyond 95% and are not eligible to perform normal operations as shown in Fig. 22.

## V. CONCLUSION AND FUTURE WORKS

This paper proposes a knowledge-based RDF personalize graph generation technique that extracts custom tuples dataset from a continuous streaming semantic repository of green-cloud-based smart grid. The proposed approach discusses peak and off block personalize graphs along with possible error exceptions and extraction of threshold factors that affects the lifespan and energy consumption of a smart meter. The presented approach delivers an effective filter that identifies the disorder elements within the range of threshold factors of lifespan and energy consumption. Also, it allocates smart meters to respective slot depending on their usage statistics to provide an optimal performance. The experimental evaluation depicts that personalize graph approach obtains an overall 72% enhancement in lifespan and 21% minimization in energy consumption by smart meters of green cloud-based smart grid. In the future, we will focus to work over effective data retrieval and compression techniques for smart meters in a green-cloud-based smart grid.

## REFERENCES

[1] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, Jan. 2015.

[2] J. W. Rittinghouse and J. F. Ransome, *Cloud Computing: Implementation, Management, and Security*. Boca Raton, FL, USA: CRC Press, 2016.

[3] S. Becker, G. Brataas, and S. Lehrig, *Engineering Scalable, Elastic, and Cost-Efficient Cloud Computing Applications: The CloudScale Method*. Springer, 2017. [Online]. Available: https://books.google.co.kr/books?hl=en&lr=&id=cyYmDwAAQBAJ&oi=fnd&pg=PR5&dq=Scalable,+Elastic,+and+Cost-Efficient+Cloud+Computing+Applications:+The+CloudScale+Method+Springer,+2017.&ots=9ra7rakXU5&sig=3aTDgNjDoXIQJAbno3nDvgjGV5Q

[4] V. A. A. Quirita *et al.*, "A new cloud computing architecture for the classification of remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 409–416, Feb. 2017.

[5] A. L. Di Salvo, F. Agostinho, C. M. Almeida, and B. F. Giannetti, "Can cloud computing be labeled as 'green'? Insights under an environmental accounting perspective," *Renew. Sustain. Energy Rev.*, vol. 69, pp. 514–526, Mar. 2017.

[6] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gen. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[7] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[8] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, Jan./Feb. 2010.

[9] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 446–464, 1st Quart., 2017.

[10] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "Iot-based big data storage systems in cloud computing: Perspectives and challenges," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, Feb. 2017.

[11] S. Zhou and M. A. Brown, "Smart meter deployment in Europe: A comparative case study on the impacts of national policy schemes," *J. Cleaner Prod.*, vol. 144, pp. 22–32, Feb. 2017.

[12] X. Su, H. Zhang, J. Riekki, A. Keränen, J. K. Nurminen, and L. Du, "Connecting IoT sensors to knowledge-based systems by transforming senML to RDF," *Procedia Comput. Sci.*, vol. 32, pp. 215–222, Jan. 2014.

[13] H. March, Á.-F. Morote, A.-M. Rico, and D. Saurí, "Household smart water metering in Spain: Insights from the experience of remote meter reading in alicante," *Sustainability*, vol. 9, no. 4, p. 582, Apr. 2017.

[14] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *Int. J. Elect. Power Energy Syst.*, vol. 63, pp. 473–484, Dec. 2014.

[15] I. S. Bayram and T. S. Ustun, "A survey on behind the meter energy management systems in smart grid," *Renew. Sustain. Energy Rev.*, vol. 72, pp. 1208–1232, May 2017.

[16] M. M. Albu, M. Sănduleac, and C. Stănescu, "Syncretic use of smart meters for power quality monitoring in emerging networks," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 485–492, Jan. 2017.

[17] M. Kuzlu, M. Pipattanasompom, and S. Rahman, "A comprehensive review of smart grid related standards and protocols," in *Proc. 5th Int. Istanbul Smart Grid Cities Congr. Fair (ICSG)*. Apr. 2017, pp. 12–16.

[18] Y. Yoldaş, A. Önen, S. Muyeen, A. V. Vasilakos, and İ. Alan, "Enhancing smart grid with microgrids: Challenges and opportunities," *Renew. Sustain. Energy Rev.*, vol. 72, pp. 205–214, May 2017.

[19] K. Kim and S.-I. Jin, "Branch-based centralized data collection for smart grids using wireless sensor networks," *Sensors*, vol. 15, no. 5, pp. 11854–11872, 2015.

[20] A. Crapo, X. Wang, J. Lizzi, and R. Larson, "The semantically enabled smart grid," *Road Interoperable Grid (Grid-Interop)*, to be published. [Online]. Available: https://pdfs.semanticscholar.org/940e/1d2bdffff3a6566a5ab8618b907dace51fbf.pdf

[21] A. Pena and Y. K. Penya, "Distributed semantic repositories in smart grids," in *Proc. 9th IEEE Int. Conf. Ind. Inform. (INDIN)*, Apr. 2011, pp. 721–726.

[22] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective," *Proc. IEEE*, vol. 104, no. 5, pp. 1058–1070, May 2016.

[23] Y. K. Penya, J. C. Nieves, A. Espinoza, C. E. Borges, A. Peña, and M. Ortega, "Distributed semantic architecture for smart grids," *Energies*, vol. 5, no. 11, pp. 4824–4843, 2012.

[24] T. H. Nguyen, V. Nunavath, and A. Prinz, "Big data metadata management in smart grids," in *Big Data Internet Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 189–214. [Online]. Available: https://www.researchgate.net/profile/Marta_Jablonska/publication/261793787_Internet_of_things_in_smart_grid_deployment/links/0a85e5357ac9d6bcef000000.pdf

[25] M. R. Jabłońska, "Internet of Things in smart grid deployment," *Rynek Energii*, vol. 2, p. 111, Apr. 2004.

[26] J. S. Zieliński, "Internet of Everything (IoE) in smart grid," *Przegląd Elektrotechniczny*, vol. 91, no. 3, pp. 157–159, 2015.

[27] S. Haben, C. Singleton, and P. Grindrod, "Analysis and clustering of residential customers energy behavioral demand using smart meter data," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 136–144, Jan. 2016.

[28] J. Every, L. Li, and D. G. Dorrell, "Leveraging smart meter data for economic optimization of residential photovoltaics under existing tariff structures and incentive schemes," *Appl. Energy*, vol. 201, pp. 158–173, Sep. 2017.

[29] Y. K. Penya, O. Kamara, and A. Pena, "IEC60870 meter smart SOA management," in *Proc. IEEE PES Innov. Smart Grid Technol.*, Nov. 2011, pp. 1–7.

[30] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 28–37, 2001.

[31] A. Wagner, S. Speiser, and A. Harth, "Semantic Web technologies for a smart energy grid: Requirements and challenges," in *Proc. Int. Conf. Posters Demonstrations Track*, vol. 658. 2010, pp. 33–36.

[32] A. Owens *et al.*, "Clustered TDB: A clustered triple store for jena," in *Proc. WWW*, Madrid, Spain, Apr. 2009, pp. 20–24.

[33] S. Mayer, J. Hodges, D. Yu, M. Kritzler, and F. Michahelles, "An open semantic framework for the industrial Internet of Things," *IEEE Intell. Syst.*, vol. 32, no. 1, pp. 96–101, Jan. 2017.

[34] R. De Virgilio, "A linear algebra technique for (de) centralized processing of SPARQL queries," in *Proc. Int. Conf. Conceptual Modeling*, 2012, pp. 463–476.

[35] R. De Virgilio, "Smart rdf data storage in graph databases," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., CCGrid*, Sep. 2017, pp. 872–881.

[36] A. E. Wegner, L. Ospina-Forero, R. E. Gaunt, C. M. Deane, and G. Reinert. (2017). "Identifying networks with common organizational principles." [Online]. Available: https://arxiv.org/abs/1704.00387

[37] N. Shervashidze, P. Schweitzer, E. J. V. Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler–Lehman graph kernels," *J. Mach. Learn. Res.*, vol. 12, pp. 2539–2561, Sep. 2011.

[38] G. K. de Vries, "A fast approximation of the weisfeiler-lehman graph kernel for rdf data," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 606–621.

[39] K. Kim, B. Moon, and H.-J. Kim, "R$_3$F: RDF triple filtering method for efficient SPARQL query processing," *World Wide Web*, vol. 18, no. 2, pp. 317–357, 2015.

[40] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. North Chelmsford, MA, USA: Courier Corporation, 1998.

[41] M. La Scala, S. Bruno, C. A. Nucci, S. Lamonaca, and U. Stecchi, *From Smart Grids to Smart Cities: New Challenges in Optimizing Energy Grids*. Hoboken, NJ, USA: Wiley, 2017.

[42] S. Bhela, V. Kekatos, and S. Veeramachaneni, "Enhancing observability in distribution grids using smart meter data," *IEEE Trans. Smart Grid*, to be published, doi: 10.1109/TSG.2017.2699939.

[43] R. Jonker and T. Volgenant, "Improving the Hungarian assignment algorithm," *Oper. Res. Lett.*, vol. 5, no. 4, pp. 171–175, 1986.

[44] (2017). *Apache Jena—TDB*. [Online]. Available: https://jena.apache.org/documentation/tdb/

[45] T. Weibel. (2017). *Smart Meter Peak and off Load Datasets*. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart

[46] (2017). *Apache Hadoop!*. [Online]. Available: http://hadoop.apache.org/

**ISMA FARAH SIDDIQUI** received the B.E. degree in software engineering in 2006 and the M.E. degree in information technology from the Mehran University of Engineering and Technology, Pakistan, in 2008. She is currently pursuing the Ph.D. with the Department of Computer Science and Engineering, Hanyang University ERICA, South Korea, funded by the Higher Education Commission, Pakistan. Since 2006, she has been with the Department of Software Engineering, Mehran University of Engineering and Technology, where she is currently an Assistant Professor. Her research interests include smart environment, semantic web, IoT, green cloud, and big data.

**SCOTT UK-JIN LEE** received the B.E. degree in software engineering and the Ph.D. degree in computer science from the University of Auckland, New Zealand. He was as Post-Doctoral Research Fellow with Commissariat à l'énergieatomique et aux énergies alternatives, France. He has been with the Department of Computer Science and Engineering, Hanyang University ERICA, South Korea, since 2011. He is the Head of Department of Software for Emerging Technology major. His research interests include software engineering, formal methods, Web, and IoT. He is also a member of the Korean Institute of Information Scientists and Engineers and the Korean Society of Computer and Information. He has served as an editor, the technical chair and a committee member of several journals and conferences.

**ASAD ABBAS** received the B.S. degree information technology from the University of the Punjab, Pakistan, in 2011. He Joined the MS-Leading to Ph.D. Program with the Department of Computer Science and Engineering, Hanyang University ERICA, Ansan, South Korea, funded by the Higher Education Commission of Pakistan in 2014. His research interests include software product line, software requirement traceability, and IoT applications.

**ALI KASHIF BASHIR** (M'15–SM'16) received the Ph.D. degree in computer science and engineering from Korea University, South Korea. He is currently an Associate Professor of information and communication technologies with the Faculty of Science and Technology, University of the Faroe Islands, Faroe Islands. He held appointments with the Nara National College of Technology, Japan, the National Fusion Research Institute, South Korea, Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. In 2013, he joined the Graduate School of Information Science and Technology, Osaka University. His research interests include cloud computing, NFV/SDN, network virtualization, IoT, computer networks, RFID, sensor networks, wireless networks, and distributed computing. He is serving as the Editor-in-chief of the IEEE Internet Technology Policy Newsletter and the IEEE Future Directions Newsletter. He is an Editorial Board Member of journals, such as the IEEE Access, the *Journal of Sensor Networks*, and the *Data Communications*. He has chaired several conference sessions, gave several invited and keynote talks, and reviewed the technology leading articles for journals, such as the IEEE Transactions on Industrial Informatics, the *IEEE Communication Magazine*, the IEEE Communication Letters, and the IEICE Journals, and conferences, such as the IEEE Infocom, the IEEE ICC, the IEEE Globecom, and the IEEE Cloud of Things.